

SPECIFICATION

Electronic Version 1.2.8

Stylesheet Version 1.0

METHODS AND APPARATUS FOR SECURE PROGRAMMING OF AN ELECTRICITY METER

Cross Reference to Related Applications

This application claims the benefit of U.S. Provisional Application Number 60/221,634 filed July 28, 2000.

Background of Invention

[0001] This invention relates generally to electronic electricity meters, and more particularly to methods and apparatus for preserving the security of programmable electronic electricity meters.

[0002] Modern digital electronic electricity meters require a high level of security of operation to prevent theft of service by meter tampering, including the possibility of tampering with the meter data or with the many programmable parameters in the meter which control meter operation and can significantly effect the billing quantities calculated by the meter. It has been a practice in some areas to control the changing of programmed meter parameters by preventing or severely constraining these changes. In practice this process is very onerous, as there are valid reasons to revise these parameters from time to time.

[0003] In at least one known type of electronic electricity meter, unauthorized program changes have been prevented by use of a feature known as a seal flag. Once a meter has been programmed by an authorized agent the seal flag feature is set, preventing further changes to controlled parameters. The meter design incorporates the selection of which tables are sealed and which are not. Although these meters provide security from unauthorized program changes, they lack programming flexibility in those cases in which there is a legitimate need to modify metering parameters.

[0004] In at least one known type of electronic electricity meter, a record of the date last programmed and a programmer ID is kept. It is also known to use tables to log event histories in meters. These meters provide the flexibility needed to modify metering parameters. However, there exists the possibility of unauthorized parties tampering with the log files themselves.

Summary of Invention

[0005] In one aspect, a method for creating a secure program history log for a programmable device including a microprocessor, at least one communications port for communicating with the microprocessor and at least one memory device electrically connected to the microprocessor and including a program history log is provided. The method comprises communicating program parameters to the microprocessor, creating a log entry utilizing the microprocessor and the program parameters, and writing the log entry into the program history log utilizing the microprocessor.

[0006] In another aspect, a system for creating a secure program history log for a programmable device is provided. The system comprises at least one communications port, said communications port configured to receive inputs comprising program parameters, a microprocessor configured to receive said program parameters from said communications port and create a log entry based on said program parameters, and at least one memory device electrically connected to said microprocessor and comprising said program history log, said microprocessor further configured to write said log entry into said program history log, thereby protecting said program history log from manipulation via direct communication from said communications port.

[0007] In still another aspect, an electronic electricity meter is provided that comprises a communications port, said communications port configured to receive meter input parameters, a microprocessor configured to receive said meter input parameters from said communications port and determine energy consumption based upon said meter input parameters, said microprocessor further configured to create a program history log entry when meter parameters are received, and at least one memory device electrically connected to said microprocessor and comprising a program history log, said microprocessor further configured to write said log entry into said program history log.

[0008] In yet another aspect, an electronic electricity meter comprising a microprocessor, at least one memory device and a communications port is provided. The microprocessor is configured to determine energy consumption based upon at least one meter input parameter. The at least one memory device is electrically connected to said microprocessor and comprises a program history log. The communications port is configured to receive said at least one meter input parameter for use by said microprocessor. The microprocessor is configured to create a program history log entry and configured to write said log entry into said program history log when said at least one meter parameter is received, said program history log comprising at least one of an entry sequence number, a transaction number, a date and time stamp, and a table identifier.

Brief Description of Drawings

[0009] Figure 1 is a block diagram of one embodiment of an electronic electricity meter of the present invention.

[0010] Figure 2 is a representation of a program history log for the electronic electricity meter shown in Figure 1 comprising identifiable types of records within a more general table.

[0011] Figure 3 is a representation of a remote computer system in communication via a telephone modem with the meter shown in Figure 1.

Detailed Description

[0012] In one embodiment of the present invention, and referring to Figures 1 and 2, a programmable electronic electricity meter 10 is provided with a memory 12 for maintaining a program history log 14. Meter 10, in accordance with known meters, includes voltage sensors and current sensors coupled to a power supply of a metered electrical system, such as a residence, business, or other establishment wherein power consumption is monitored by meter 10. In one embodiment, the voltage and current sensors are coupled to an analog-to-digital (A/D) converter that converts signals from the voltage and current sensors to digital output signals to meter electronics.

[0013] Meter electronics accumulate data and determine energy consumption and quantities of interest according to known methods and techniques. Data outputs are determined according to programmed parameters input to the electronics. In an effort to monitor accuracy of meter parameters, program history log 14 is included to determine a state of meter parameters,

changes to those parameters, and information relating to changed or altered meter parameters. Unauthorized changes may therefore be detected and redressed to thwart unauthorized meter programming and tampering that illegitimately affects billed quantities for energy consumption.

[0014] In one embodiment, program history log 14 comprises entries or records 16 of an identifiable type. In another embodiment, program history log comprises a more general event or security program history log 18 (shown in Figure 2), which includes records 16 of an identifiable type and other general event records 20. In yet another embodiment, program history log 14 is contained in a separate memory element 22 within meter 10. In each of these embodiments, parameters of meter 10, including but not limited to, for example, selections of quantities for load profiles, real-time pricing schedules, and time of use metering mode parameters are programmed into meter 10. Programming of meter parameters is performed, for example, via an optical communications port 24, a telephone modem 26, an RS-232 port 28, or another communication port (not shown) of meter 10 according to known methods and techniques. The various programmable parameters are stored in tables 30 in system memory, such as memory 22. As contemplated herein, a single programmable parameter stored in memory that is not part of a larger table 30 may be considered as being stored in a table 30 having a single parameter.

[0015] Each programming session with meter 10 creates or generates a log entry 16 in program history log 14. Thus, remote programming sessions of meter 10 taking place via modem 26 create a log entry 16, as do programming sessions that take place via optical port 24 of meter 10 or via other input ports of meter 10. Each log entry 16 includes information relating to programming of meter 10 so that changes to meter parameters may be identified, and, if necessary, such as for unauthorized changes or tampering with meter 10, meter parameters may be restored and further, log entry 16 may provide information that may reveal the perpetrator of programming changes.

[0016] For example, in an illustrative embodiment, each log entry 16 includes an entry sequence number 32, a date and time stamp 34, a transaction number 36 for the programming event, and an entry 38 identifying each table 30 written for the logged event. A meter identifier 40 is also stored in conjunction with the history log 14, in a designated location, such as, for example, a header section 42 of table memory or as part of each entry 16. Each log entry 16 is

written into program history log 14 by a meter processor 44, which as noted above accumulates metered data quantities, makes calculations and otherwise manipulates data inputs into usable output form.

[0017] In one embodiment, memory 12 (or memory 22 in another embodiment) into which a log entry 16 is stored is configured to be accessible internally by processor 44 of electricity meter 10 to the exclusion of external inputs. As such, data entry into meter 10 via communication ports 24, 26, 28 cannot be directly written into history log 14 via communication from devices external to meter 10. Consequently, meter parameter data may not be easily altered with conventional techniques, including but not limited to specific memory addressing. Thus, program history log 14 may generally not be cleared or erased, and log entries 16 may not be changed or altered once log 16 is written into program history log 14 by processor 44. It is contemplated, however, that in a further embodiment, a "master reset" function is provided that can be accessed for disaster recovery of meter 10 that would wholly or partially reset data values of program history log 14.

[0018] As noted above, log entries 16 are written or added to program history log 14 as programming events occur. In one embodiment, when program history log 14 is full, i.e., when all available designated memory locations are written to, log entries 16 wrap around as though program history log 16 were a circular buffer, so that an oldest entry is overwritten by a newest entry. Thus, for the embodiment illustrated in Figure 1, program history log 14 of memory 12 contains seven memory locations for storing seven log entries relating to seven respective programming events. Upon the occurrence of the eighth programming event, one of the seven memory locations (and in a particular embodiment the memory location containing a log entry 16 relating to the first or earliest recorded programming event) is written over and replaced by a log entry containing information relating to the eighth programming event. Successive programming events likewise are written to the remaining memory locations and overwrite prior data therein.

[0019] In a further embodiment, when program history log 14 is filled with log entries 16 that have not been read externally, no further programming changes are allowed until program history log 14, and more specifically, log entries 16 are read out. In this embodiment, meter 10 maintains a pointer, inaccessible external to, or from outside of, meter 10. The pointer points to the last log entry 16 read out from the meter.

[0020] In one embodiment, program history log 14 is read externally through one of communications ports 24, 26, or 28 of meter 10 with known devices according to known techniques. In particular embodiments, either an entire history log 14 is read out from meter 10, or specific entries 16 made since the last read out, as indicated by the pointer referred to above, are read out from meter 10.

[0021] In an exemplary embodiment, and referring now to Figure 3, new values of parameters programmed into meter 10 are stored in a computer system 46 outside of, or external to, 10. When a meter programming event occurs, the programming process is controlled by computer 46, which executes programming software 48 for programming meter 10. Programming software 48 creates a record 50 for each programming event of meter 10. Record 50 is then stored in a file 52. Record 50 includes information in meter program history log 14, or more specifically in log entries 16, that are created for the programming event, plus the new values of the parameters programmed in the session. In one embodiment, programming software 48 creates record 50 by reading meter program tables 30 (shown in Figure 1) after the programming event. In another embodiment, an even more robust record is created by taking a before reading, i.e., a reading of tables 30 before the programming event, and including the "before" readings in record 50. Suitable security measures for program history records 50 are provided at computer 46, such as password security or file security known in the art. In one embodiment, file contents including records 50 are written to a write-once medium 54, such as a non-reprogrammable programmable read only memory, or a non-rewritable CD-ROM.

[0022] In one embodiment in which multiple computer devices 46 are used to program various meters 10 in a utility system, program history files of records 50 are returned to a central computer system 56 for consolidation, much like conventional meter readings are now returned.

[0023] The use of entry sequence numbers 32 facilitates easy validation and manipulation of entries. Entry transaction numbers 36 are generated inside the meter by a process providing an output that is difficult to predict, such as an encryption algorithm or a hashing algorithm. Entry transaction numbers 36 are keys that provide secure identifiers linking entries 18 in the internal program history log 14 of a meter 10 to the record 48 of parameters programmed by the session in the computer file 50.

[0024] It will thus be recognized that the above-described embodiments provide a secure program

